

Automazione

17 marzo 2022

Esercizio 1

Si consideri un sistema di automazione di una azienda di logistica in cui, a livello di coordinamento, è necessario gestire i seguenti task periodici:

- ogni 4 t.u. viene effettuata l'azione di riempimento dei pacchi, impiegando 1 t.u.;
- ogni 9 t.u. viene effettuata l'azione di imballaggio dei pacchi in 3 t.u.;
- ogni 12 t.u. viene eseguita l'azione di smistamento dei pacchi in 4 t.u.

C'è un addizionale task aperiodico di controllo della qualità, che ha un activation time 2 t.u., una deadline relativa di 30 t.u. e un computation time di 3 t.u. Si ipotizzi che tutti i task descritti siano indipendenti, dal punto di vista funzionale, l'uno dall'altro. I task periodici devono essere gestiti con una modalità di scheduling hard real time.

1. Verificare se sussiste la condizione necessaria di schedulabilità dei task periodici.
2. Verificare se sussiste almeno una delle condizioni sufficienti di schedulabilità, utilizzando l'algoritmo RMPO.
3. Eseguire lo scheduling RMPO. Nel caso in cui RMPO non sia in grado di schedulare i task periodici in maniera hard real time, eseguire lo scheduling utilizzando l'algoritmo EDF.
4. Utilizzando l'algoritmo di scheduling scelto nel punto precedente, verificare se il task aperiodico riesce a essere eseguito in maniera hard real time utilizzando:
 - a) uno scheduling in background;
 - b) oppure, un processo polling server caratterizzato da $T_{SRV} = 12$ t.u. e $C_{SRV} = 1$ t.u.;
 - c) oppure, un processo deferring server caratterizzato da $T_{SRV} = 12$ t.u. e $C_{SRV} = 1$ t.u.

Esercizio 2

Si consideri un motore master (con posizione angolare $\theta(t)$) e un motore slave (con posizione angolare $q_1(t)$). Sull'asse del motore slave è posto un riduttore con rapporto di riduzione 12:1. Il riduttore è solidale a una puleggia circolare di raggio $R = 10$ [cm] che muove un nastro trasportatore. Si vuole imporre una *sincronizzazione* del moto $q_1(\theta)$ tra i motori master e slave, espressa da un polinomio di terzo grado tale per cui, quando la posizione angolare $\theta(t)$ del master ha l'andamento riportato in Fig. 1, un generico carico presente sul nastro (la cui posizione lineare è indicata con $q_2(t)$) si muova da 0 a 30 [cm], partendo da fermo e arrivando con velocità pari a 0.5 [m/s]. Determinare i valori numerici dei coefficienti del polinomio cubico $q_1(\theta)$ (con le unità di misura).

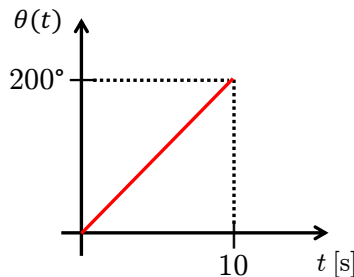


Figura 1: Andamento temporale della posizione angolare del motore master

Esercizio 3

Per la rete di Petri mostrata in Fig. 2, si calcolino tutti i P-invarianti canonici a supporto minimo e tutti i T-invarianti. Si studino quindi le proprietà di vivezza, limitatezza e reversibilità. Nel caso in cui una o più di tali proprietà non sia soddisfatta, progettare un supervisore che le garantisca tutte. Costruire l'albero di raggiungibilità della rete finale (eventualmente supervisionata).

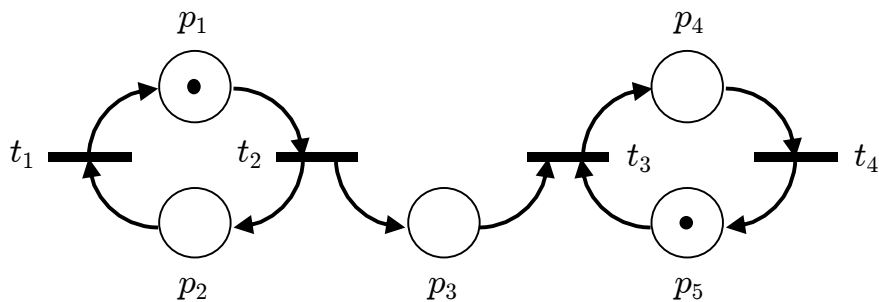


Figura 2: La rete di Petri PN da analizzare.

[150 minuti (2.5 ore); libri aperti]

Soluzione

17 marzo 2022

Esercizio 1

1. Per verificare la condizione necessaria si calcola il fattore di utilizzazione dei task periodici hard real time:

$$U = \frac{1}{4} + \frac{3}{9} + \frac{4}{12} = \frac{9 + 12 + 12}{36} = \frac{33}{36} \simeq 0.92.$$

2. Verificata la condizione necessaria, controlliamo se sussiste almeno una condizione sufficiente.

$$U_{lsm}(RMPO) = n \left(2^{1/n} - 1 \right) = 3 \left(2^{1/3} - 1 \right) \simeq 0.78.$$

Dato che $U > U_{lsm}$, questa condizione sufficiente non è verificata. Inoltre i 3 task non sono legati tra loro da relazioni armoniche. Non possiamo quindi dire a priori se RMPO è in grado di schedulare i task.

3. La soluzione dello scheduling RMPO è riportata in Fig. 3. Da questa si evince che RMPO non è in grado di schedulare in maniera hard real time i task periodici come richiesto, in quanto la prima occorrenza del task A_3 non è terminata quando cede il passo alla quarta occorrenza del task A_1 . Pertanto, passiamo a schedulare il problema dato usando EDF. La soluzione dello scheduling EDF, con i soli task periodici, è riportata in Fig. 4. Come atteso, EDF è una soluzione del problema dato e quindi verrà selezionato come algoritmo per procedere nei passi successivi.

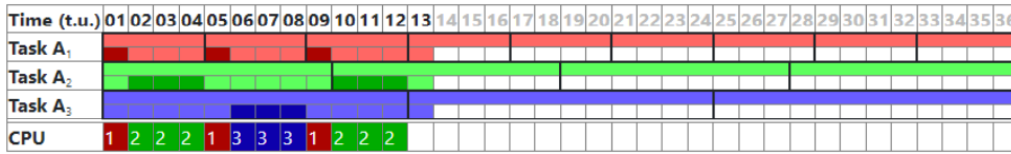


Figura 3: Scheduling RMPO.

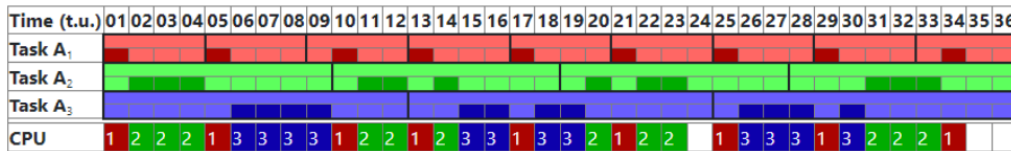


Figura 4: Scheduling EDF.

4.a) Aggiungiamo il task aperiodico (A_4) utilizzando una politica in background (Fig. 5). Il task aperiodico non riesce ad essere eseguito in tempo se si utilizza una politica in background, in quanto le sue ultime 2 t.u. vengono eseguite oltre la deadline assoluta di 32 t.u.

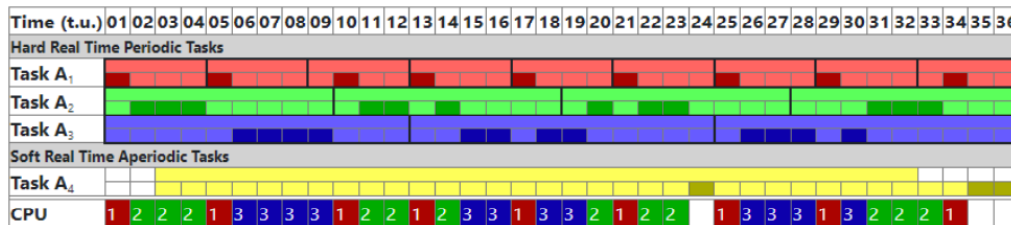


Figura 5: Scheduling EDF del task aperiodico A_4 con una politica in background.

4.b) Aggiungendo il processo server (A_{SRV}), il coefficiente di utilizzazione diventa:

$$U = \frac{1}{4} + \frac{3}{9} + \frac{4}{12} + \frac{1}{12} = \frac{9 + 12 + 12 + 3}{36} = \frac{36}{36} = 1.$$

Verificata la condizione necessaria e sufficiente per EDF, la soluzione dello scheduling con un processo polling server è riportata in Fig. 6. Da questa segue che l'ultima t.u. del task aperiodico non viene eseguita entro la deadline assoluta. Si noti inoltre che le istanze dei task A_3 e A_{SRV} hanno la stessa priorità; pertanto, la scelta di quale task servire per primo è appannaggio dell'implementazione dell'algoritmo. In particolare, nella soluzione mostrata, viene data precedenza alle istanze dei task periodici. Ma le considerazioni sul task aperiodico non cambiano.

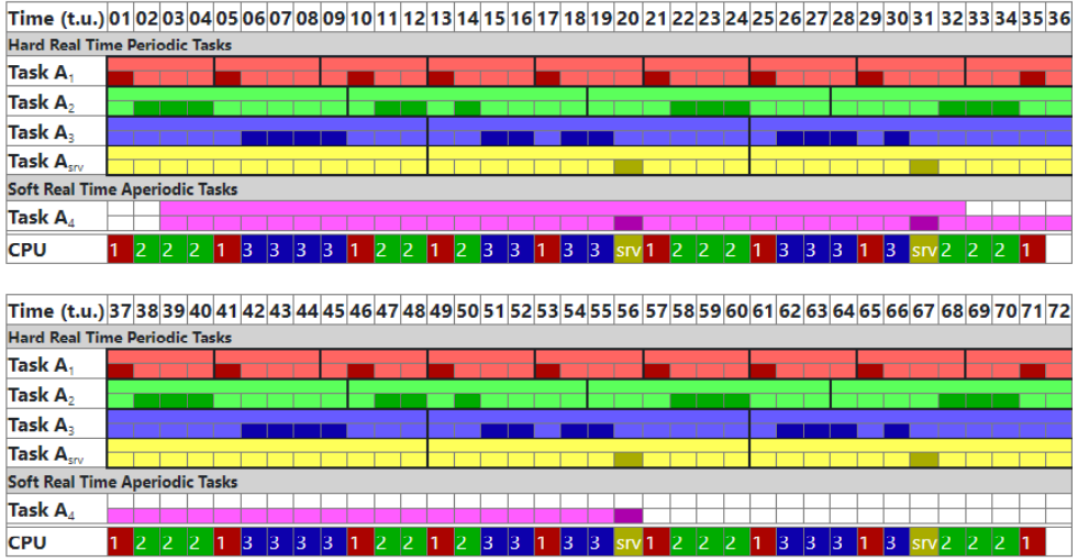


Figura 6: Scheduling EDF del task aperiodico A_4 con un processo polling server.

4.c) Se si utilizza infine una politica deferring server (Fig. 7), si conclude che le 3 t.u. del task aperiodico possono essere eseguite entro la deadline assoluta. Anche in questo caso, si noti che le istanze dei task A_3 e A_{SRV} hanno la stessa priorità; pertanto, la scelta di quale task servire per primo è appannaggio dell'implementazione dell'algoritmo. Nella soluzione mostrata viene data precedenza alle istanze dei task periodici, ma le considerazioni sul task aperiodico non cambiano.

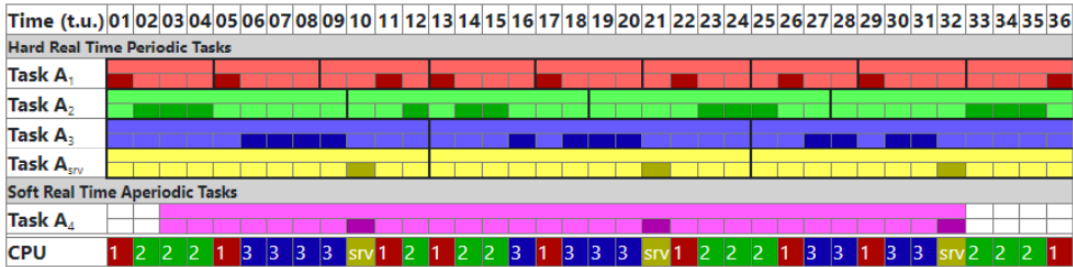


Figura 7: Scheduling EDF del task aperiodico A_4 con un processo deferring server.

Esercizio 2

Il moto nel tempo del motore master, come rappresentato in Fig. 1, è descritto dalla

$$\theta(t) = \frac{\Delta}{T} t = \frac{200^\circ \cdot (\pi/180^\circ)}{T} t \simeq \frac{3.49}{T} t = 0.349 t \text{ [rad]}, \quad t \in [0, T] = [0, 10] \text{ [s]},$$

con velocità costante $\dot{\theta} = \Delta/T = 0.349 \text{ [rad/s]}$. Attraverso il riduttore e la puleggia, il legame (proporzionale) tra il moto del motore slave e quello del carico è dato da

$$q_1(t) = N_r \frac{q_2(t)}{R} = \frac{12}{0.1} q_2(t) = 120 q_2(t) \text{ [rad]}, \quad t \in [0, T] = [0, 10] \text{ [s]}.$$

Le condizioni temporali al contorno per il moto del carico sono

$$q_2(0) = 0, \quad \dot{q}_2(0) = 0, \quad q_2(T) = 0.3 \text{ [m]}, \quad \dot{q}_2(T) = 0.5 \text{ [m/s]},$$

che si traducono in quelle per il motore slave

$$q_1(0) = 0, \quad \dot{q}_1(0) = 0, \quad q_1(T) = 36 \text{ [rad]}, \quad \dot{q}_1(T) = 60 \text{ [rad/s]}.$$

Per sincronizzare la posizione del motore slave a quella del motore master, si utilizza qui una legge parametrica $q_1(\theta)$ espressa da un polinomio cubico della forma

$$q_1(\theta) = a_0 + a_1\theta + a_2\theta^2 + a_3\theta^3, \quad \theta \in [0, \Delta]. \quad (1)$$

Si noti anzitutto il legame tra derivate temporali e geometriche

$$\dot{q}_1(t) = \frac{dq_1(t)}{dt} = \frac{dq_1(\theta)}{d\theta} \frac{d\theta(t)}{dt} = q'_1(\theta) \dot{\theta}(t) \quad \Rightarrow \quad q'_1(\theta) = \frac{\dot{q}_1(t)}{\dot{\theta}(t)}, \quad \text{in ogni } \theta = \theta(t).$$

Per ricavare i coefficienti incogniti nella (1), dobbiamo quindi imporre le seguenti condizioni al contorno (nei valori indicati per gli argomenti):

$$\begin{aligned} q_1(\theta = 0) &= q_1(t = 0) &\Rightarrow & a_0 = 0 \\ q'_1(\theta = 0) &= \frac{\dot{q}_1(t = 0)}{\dot{\theta}(t = 0)} &\Rightarrow & a_1 = \frac{0}{0.349} = 0 \\ q_1(\theta = \Delta) &= q_1(t = T) &\Rightarrow & a_2\Delta^2 + a_3\Delta^3 = 36 \\ q'_1(\theta = \Delta) &= \frac{\dot{q}_1(t = T)}{\dot{\theta}(t = T)} &\Rightarrow & 2a_2\Delta + 3a_3\Delta^2 = \frac{60}{0.349}. \end{aligned}$$

Risolvendo le ultime due condizioni rispetto alle due incognite residue a_2 (in $[\text{rad}^{-1}]$) e a_3 (in $[\text{rad}^{-2}]$), si ottiene per la (1)

$$q_1(\theta) = -40.3785 \theta^2 + 12.4140 \theta^3, \quad \theta \in [0, \Delta] = [0, 3.49]. \quad (2)$$

Nelle Figg. 8–9 si riportano gli andamenti di interesse delle grandezze temporali dei due motori, ossia le posizioni $\theta(t)$ e $q_1(t)$ e le velocità $\dot{\theta}(t)$ e $\dot{q}_1(t)$, come pure il loro legame geometrico $q_1(\theta)$ con la relativa derivata prima rispetto al parametro, $q'_1(\theta) = dq_1(\theta)/d\theta$. Per una migliore lettura del passaggio da un generico istante di tempo t al valore $\theta = \theta(t)$ del motore master e poi da questo valore θ al valore $q_1 = q_1(\theta)$ del motore slave, si veda anche la costruzione in Fig. 10.

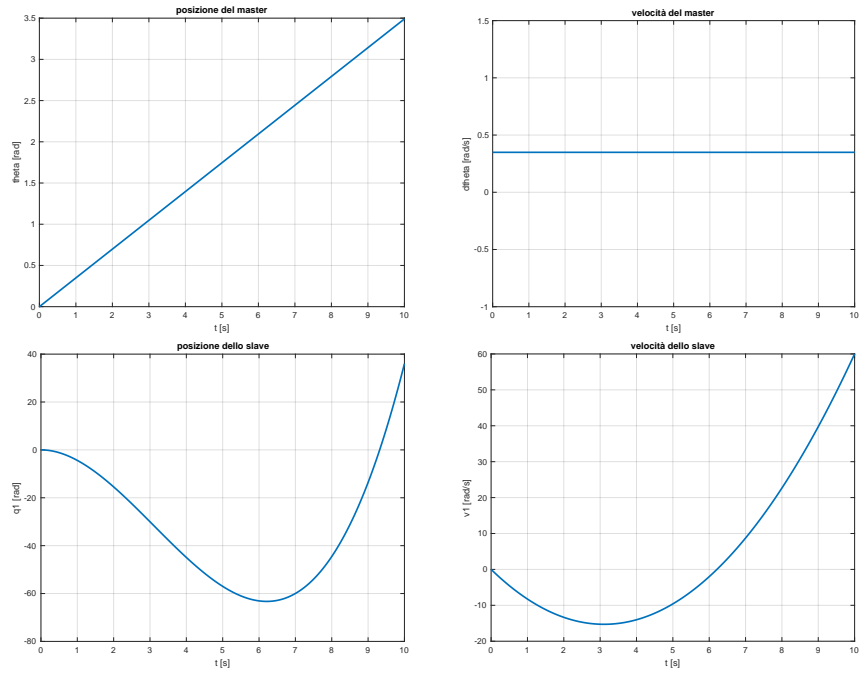


Figura 8: Posizioni e velocità nel tempo del motore master [sopra] e di quello slave [sotto].

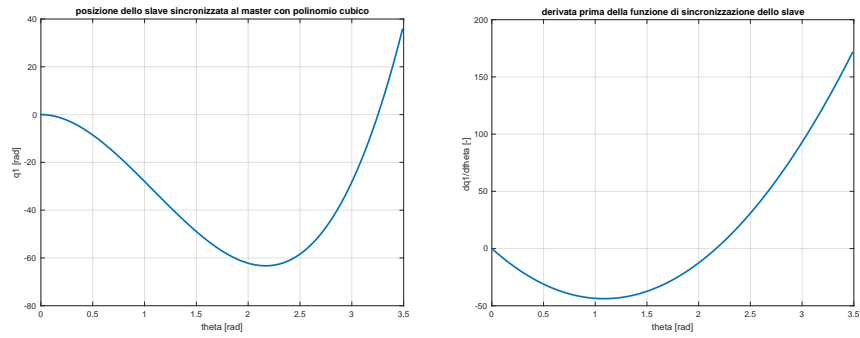


Figura 9: Sincronizzazione tra motore master e slave: posizione [a sinistra] e derivata [a destra].

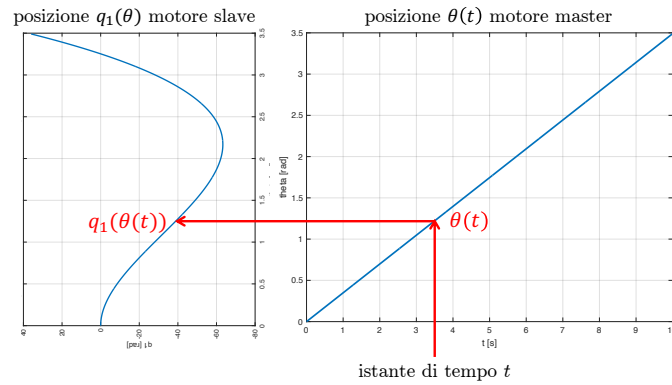


Figura 10: Lettura della sincronizzazione spaziale tra motore master e motore slave.

Esercizio 3

La rete di Petri PN considerata è di tipo ordinario (archi con pesi unitari) e rappresenta un tipico processo produttore-consumatore. Si tratta inoltre di un grafo marcato (Marked Graph): essendo presente (almeno) un token in ogni ciclo, la condizione necessaria e sufficiente per la vivezza è soddisfatta. La rete non è però limitata: è facile vedere che la sequenza di scatto $\{t_2, t_1\}^k$, con $k \in \mathbb{N}$ arbitrariamente grande, accumulerebbe nel posto p_3 un numero illimitato di token. Verifichiamo tali affermazioni con il calcolo degli invarianti.

La matrice di incidenza (5×4) C della rete e la sua marcatura iniziale \mathbf{x}_0 sono

$$C = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \quad \mathbf{x}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

E' facile verificare, ad esempio con Matlab, che il rango di C è pari a 3. Pertanto lo spazio nullo di C ha dimensione 1 e quello di C^T ha dimensione 2. Il calcolo dei P-invarianti (soluzioni non banali, intere e non negative di $C^T \boldsymbol{\gamma} = \mathbf{0}$) fornisce i due vettori canonici a supporto minimo

$$\boldsymbol{\gamma}_1^T = (1 \ 1 \ 0 \ 0 \ 0), \quad \boldsymbol{\gamma}_2^T = (0 \ 0 \ 0 \ 1 \ 1).$$

Esistono quindi due insiemi/cicli di posti in cui la somma di token si conserva durante l'evoluzione della rete: $\|I_{\gamma_1}\| = \{p_1, p_2\}$ e $\|I_{\gamma_2}\| = \{p_4, p_5\}$. In entrambi i casi, il numero di token rimane pari a quello presente in tali insiemi nella marcatura iniziale: $\boldsymbol{\gamma}_1^T \mathbf{x}_0 = 1$ (ossia il token inizialmente in p_1) e $\boldsymbol{\gamma}_2^T \mathbf{x}_0 = 1$ (ossia il token inizialmente in p_5). Il posto p_3 non è ricoperto dagli insiemi di supporto dei due P-invarianti e quindi la condizione affinché la rete sia conservativa non è valida.

Il calcolo dei T-invarianti (soluzioni intere e non negative di $C\boldsymbol{\eta} = \mathbf{0}$) fornisce l'unico vettore

$$\boldsymbol{\eta} = \alpha (1 \ 1 \ 1 \ 1)^T, \quad \alpha \in \mathbb{N}^+.$$

La condizione necessaria di reversibilità è che tutte e quattro le transizioni debbano scattare un numero uguale di volte. L'obiettivo di ottenere comunque la reversibilità dovrà essere imposto dallo stesso supervisore che, al contempo, limiterà il numero di token nella rete.

A tale scopo, imponiamo la disuguaglianza

$$\mathbf{h}^T \mathbf{x} = (0 \ 0 \ 1 \ 0 \ 0) \mathbf{x} = x_3 \leq 1 = k,$$

per tutte le marcature raggiungibili $\mathbf{x} \in \mathcal{R}\{PN\}$ della rete. La marcatura iniziale \mathbf{x}_0 soddisfa già questa disuguaglianza in quanto $\mathbf{h}^T \mathbf{x}_0 = 0$. Applicando l'algoritmo di costruzione del supervisore basato sugli invarianti, si inserisce un posto monitor p_m con una (sesta) riga aggiuntiva alla matrice di incidenza originaria C e marcatura iniziale dati da

$$C_m = -\mathbf{h}^T C = (0 \ -1 \ 1 \ 0), \quad x_{m,0} = k - \mathbf{h}^T \mathbf{x}_0 = 1 - 0 = 1.$$

La rete supervisionata è mostrata in Fig. 11.

Dall'albero di raggiungibilità in Fig. 12 (costruito con logica depth-first, espandendo in ordine lessicografico le transizioni abilitate), è possibile verificare che la rete supervisionata PN^s è limitata, viva e reversibile. L'insieme delle marcature raggiungibili $\mathbf{x}^s \in \mathcal{R}\{PN^s\}$, con $\mathbf{x}^s = (\mathbf{x}, x_m)$, contiene 8 marcature distinte (inclusa quella iniziale).

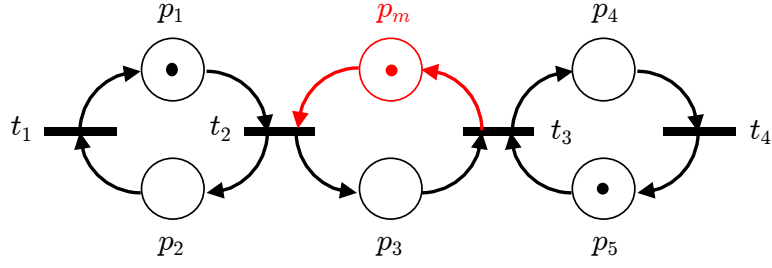


Figura 11: La rete di Petri di Fig. 2, con il posto aggiunto p_m del supervisore.

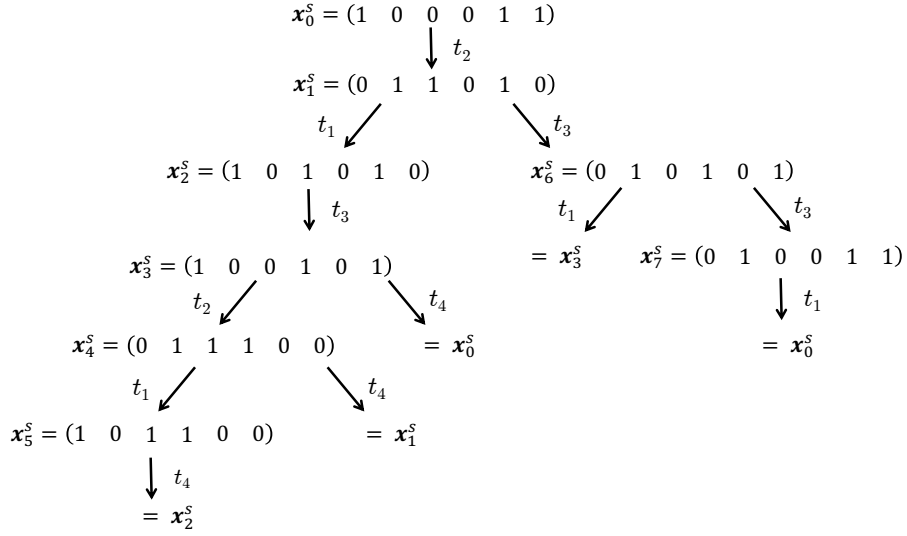


Figura 12: Albero di raggiungibilità della rete supervisionata di Fig. 11.

La proprietà di conservatività era anche deducibile direttamente dai (tre) nuovi P-invarianti della matrice di incidenza estesa. Si ha infatti:

$$\mathbf{C}^s = \begin{pmatrix} \mathbf{C} \\ \mathbf{C}_m \end{pmatrix}, \quad \boldsymbol{\gamma}^s \mathbf{C}^s = 0 \quad \Rightarrow \quad
 \begin{aligned}
 \boldsymbol{\gamma}_1^s &= (\boldsymbol{\gamma}_1^T \ 0)^T = (1 \ 1 \ 0 \ 0 \ 0 \ 0)^T \\
 \boldsymbol{\gamma}_2^s &= (\boldsymbol{\gamma}_2^T \ 0)^T = (0 \ 0 \ 0 \ 1 \ 1 \ 0)^T \\
 \boldsymbol{\gamma}_3^s &= (\mathbf{h}^T \ 1)^T = (0 \ 0 \ 1 \ 0 \ 0 \ 1)^T,
 \end{aligned}$$

da cui i sei posti della rete supervisionata sono interamente ricoperti dagli insiemi di supporto (minimo) dei P-invarianti canonici.

* * * * *